Décodage Adresse

Avant tout, il faut trouver dans l'énoncé, l'espace de stockage (le nombre de cases).

Dans l'exemple du cours :

* La mémoire EPROM et la RAM nécessitent chacune 2K octets dans l'espace d'adressage, le PIO utilise 4 octets. On suppose de plus qu'on dispose d'un espace d'adressage de 64 K octets (bus d'adresse A₀-A₁₅). Voici un choix possible

L'espace d'adressage est de 64K octets donc 64 * 1024 (le kilo en archi est égal à 1024) = $2^6 * 2^{10} = 2^{16}$

L'exposant est le nombre de cases qu'il y aura pour décoder notre adresse (ici 16, allant de A0 à A15).

Nous savons donc que nos adresses vont ressembler à un tableau de ce genre :

	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
Début																
Fin																

Dans les adresses il y a 2 parties, la partie "rouge" qui représente la partie qui ne varie pas entre le début et la fin du composant puis la partie "noire" qui elle, varie.

Pour trouver la partie "noire" il faut trouver dans l'énoncé sa taille, le nombre de cases qu'elle va occuper :

* La mémoire EPROM et la RAM nécessitent chacune 2K octets dans l'espace d'adressage , le PIO utilise 4 octets. On suppose de plus qu'on dispose d'un espace

L'EPROM et la RAM vont avoir 2K octets de cases noires, donc $2 * 1024 = 2^1 * 2^{10} = 2^{11}$

L'exposant est le nombre de cases noires (ici 11, allant de A0 à A10).

Pour le PIO c'est la même chose : 4 octets = 2^2

L'exposant est le nombre de cases noires (ici 2, allant de A0 à A1).

Nous savons que par exemple, pour l'EPROM et la RAM, le code couleur de notre adresse va ressembler à un tableau de ce genre :

	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
Début																
Fin																

Il faut maintenant remplir nos cases, chaque composant possède une adresse de début et de fin.

La partie noire va être simple à remplir, si c'est le début on met des 0 dans toutes les cases noires et si c'est la fin on met des 1.

Exemple avec la RAM:

	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
Début							0	0	0	0	0	0	0	0	0	0
Fin							1	1	1	1	1	1	1	1	1	1

Il faut maintenant remplir la partie rouge.

Comme dit précédemment, la partie rouge est la partie fixe de l'adresse du composant, donc les cases rouges du début vont être les mêmes que celles de la fin.

Mais il faut trouver le contenu des cases rouges, pour cela il faut regarder dans l'énoncé :

- mémoire EPROM au début (0) de l'espace d'adressage,
- mémoire RAM à partir de l'adresse 32K,
- -circuit d'E/S PIO dans les quatre derniers octets de l'espace d'adressage

La partie rouge de l'EPROM sera composée entièrement de 0, car il est situé au début de l'espace d'adressage du "µO" (micro ordinateur, pas un détail important).

Le tableau de l'EPROM est donc le suivant :

	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
Début	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fin	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

On passe maintenant à la RAM qui est peut-être la plus complexe à comprendre. Dans l'énoncé, il est indiqué qu'elle commence à partir de 32K $32K = 2^5 * 2^{10} = 2^{15}$

L'exposant est 15 donc en A15 il y aura un 1.

Le tableau de la RAM est donc le suivant :

	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
Début	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fin	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Enfin le PIO, on l'a vu précédemment, est composé de seulement 2 cases noires. (4 octets)

Dans l'énoncé il est dit qu'il occupe les derniers octets de l'espace d'adressage du μ O, il est donc à la fin et la partie rouge sera remplie de 1.

Le tableau du PIO est donc le suivant :

	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
Début	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
Fin	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Il faut maintenant décoder les adresses, on veut l'adresse en hexadécimal donc on découpe les adresses en groupe de 4 et on convertit.

Le résultat final :

(L'adresse en hexadécimal est à droite des tableaux)

Correction du TD décodage d'adresses															Eric Wolko			
EPROM		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0	
Adresse de début 0	début	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
Taille 2ko	fin	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FF
		32k=	=32*1	024=2	2 ⁵ *2 ¹	0=2 ¹⁵	,A ₁₅ =1	L)				ı				
RAM	début	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000
Adresse de début 32k	fin	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	87FF
PIO	début	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	FFFC
Adresse de fin : FFFF	fin	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF